

H-BEAM (형강) END BENDING VIDEO ANALYSIS SYSTEM

2024-04-29

§ Project Summary

The primary goal of this project is to set up an advanced video analysis system that monitors the end bending process in large rolling operations. This system will include video analysis servers, client monitoring PCs, video recording devices, and essential networking equipment. It aims to enhance quality control and operational efficiency by providing real-time data and insights.

§ My Contribution

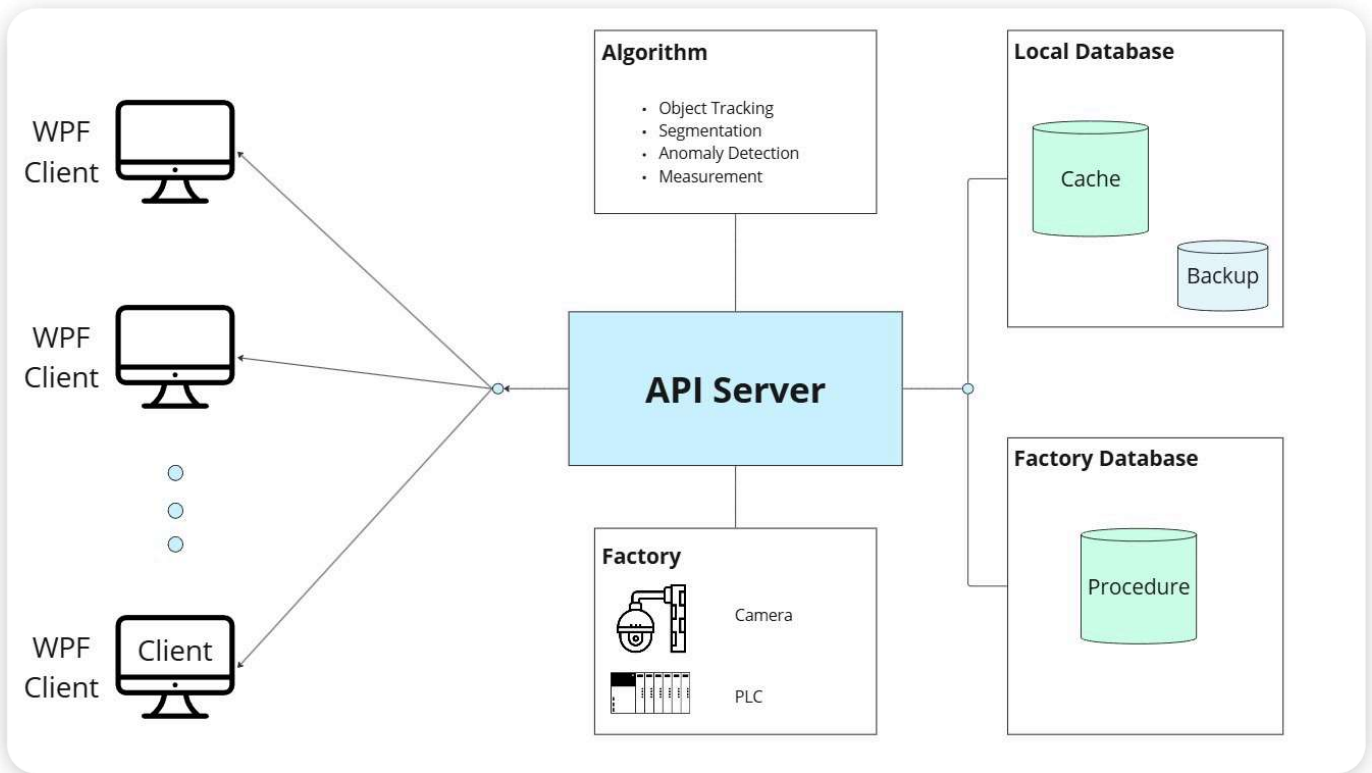
I was responsible for the end-to-end development of the system:

- **User Interface Development:** Designed and implemented a user-friendly interface using C# WPF, ensuring intuitive interaction with the system.
- **Backend Development:** Developed the server-side logic, including the integration of image processing algorithms using OpenCV, Object Tracking, Segmentation, Anomaly Detection,...
- **Database Integration:** Integrated Object Relational Mapper and MSSQL database to store and manage configuration data, including reference images and inspection results.
- **System Architecture:** Employed the MVVM (Model-View-ViewModel), Dependence Injection design pattern for a clean and maintainable codebase.
- **PLC Communication:** Developed the interface for communicating with a Programmable Logic Controller (PLC) to automate conveyor control based on inspection outcomes.

§ Software Architecture

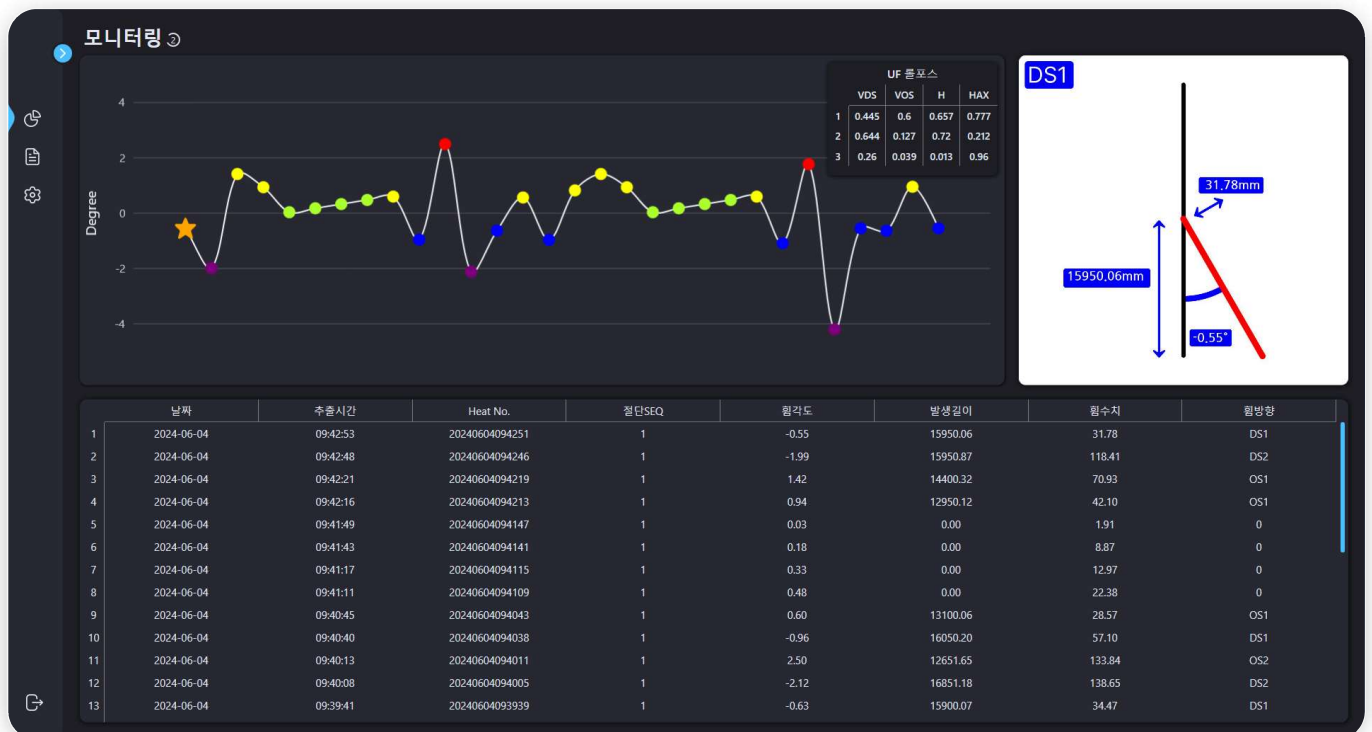


This architecture integrates multiple clients with a centralized API Server that manages data flow between advanced algorithms, local and factory databases, and real-time factory operations. The system is designed to support industrial automation and monitoring, leveraging object tracking, anomaly detection, and other advanced algorithms to enhance operational efficiency.



§ Screenshots

§ Dark Theme Interface



전체이력조회

1일 | 1월 | 2024-06-03 | 2024-06-04 | OS2,OS1,0,DS1,DS2 | 검색 | 내보내기

일련	날짜	추출시간	Heat No.	절단SEQ	휨각도	발생 길이	휨수치	휨방향
1	2024-06-04	09:42:53	20240604094251	1	-0.55	15950.06	31.78	DS1
2	2024-06-04	09:42:48	20240604094246	1	-1.99	15950.87	118.41	DS2
3	2024-06-04	09:42:21	20240604094219	1	1.42	14400.32	70.93	OS1
4	2024-06-04	09:42:16	20240604094213	1	0.94	12950.12	42.10	OS1
5	2024-06-04	09:41:49	20240604094147	1	0.03	0.00	1.91	0
6	2024-06-04	09:41:43	20240604094141	1	0.18	0.00	8.87	0
7	2024-06-04	09:41:17	20240604094115	1	0.33	0.00	12.97	0
8	2024-06-04	09:41:11	20240604094109	1	0.48	0.00	22.38	0
9	2024-06-04	09:40:45	20240604094043	1	0.60	13100.06	28.57	OS1
10	2024-06-04	09:40:40	20240604094038	1	-0.96	16050.20	57.10	DS1
11	2024-06-04	09:40:13	20240604094011	1	2.50	12651.65	133.84	OS2
12	2024-06-04	09:40:08	20240604094005	1	-2.12	16851.18	138.65	DS2
13	2024-06-04	09:39:41	20240604093939	1	-0.63	15900.07	34.47	DS1
14	2024-06-04	09:39:36	20240604093933	1	0.57	14000.05	28.13	OS1
15	2024-06-04	09:39:09	20240604093907	1	-0.96	17050.23	62.23	DS1
16	2024-06-04	09:39:04	20240604093902	1	0.82	14950.12	43.79	OS1
17	2024-06-04	09:38:37	20240604093835	1	1.42	14400.32	70.93	OS1
18	2024-06-04	09:38:32	20240604093830	1	0.94	12900.12	41.95	OS1
19	2024-06-04	09:38:05	20240604093803	1	0.03	0.00	1.91	0
20	2024-06-04	09:38:00	20240604093758	1	0.18	0.00	8.87	0
21	2024-06-04	09:37:33	20240604093731	1	0.33	0.00	12.97	0
22	2024-06-04	09:37:28	20240604093726	1	0.48	0.00	22.38	0

Totals: 7581 | 100

Pages: 76 | 1 | Goto →

설정

UI 설정

Theme: Light Dark

휨 발생 Tracking

Tracking 수: 10 20 30

서버

서버 주소:

정보

공회 영상분석 시스템
V2.0.3.2
© 2023 OCST

§ Light Theme Interface

모니터링

	VDS	VOS	H	HAX
1	0.881	0.906	0.902	0.918
2	0.082	0.727	0.012	0.104
3	0.345	0.936	0.299	0.859

날짜	추출시간	Heat No.	절단SEQ	휨각도	발생길이	휨수치	휨방향
2024-06-04	09:44:28	20240604094426	1	0.60	13100.06	28.57	OS1
2024-06-04	09:44:23	20240604094421	1	-0.96	16050.20	57.10	DS1
2024-06-04	09:43:57	20240604094355	1	2.06	12551.22	112.33	OS2
2024-06-04	09:43:52	20240604094350	1	-1.67	16050.63	100.53	DS2
2024-06-04	09:43:25	20240604094323	1	-0.63	15900.07	34.47	DS1
2024-06-04	09:43:20	20240604094318	1	0.57	14000.05	28.13	OS1
2024-06-04	09:42:53	20240604094251	1	-0.55	15950.06	31.78	DS1
2024-06-04	09:42:48	20240604094246	1	-1.99	15950.87	118.41	DS2
2024-06-04	09:42:21	20240604094219	1	1.42	14400.32	70.93	OS1
2024-06-04	09:42:16	20240604094213	1	0.94	12950.12	42.10	OS1
2024-06-04	09:41:49	20240604094147	1	0.03	0.00	1.91	0
2024-06-04	09:41:43	20240604094141	1	0.18	0.00	8.87	0
2024-06-04	09:41:17	20240604094115	1	0.33	0.00	12.97	0

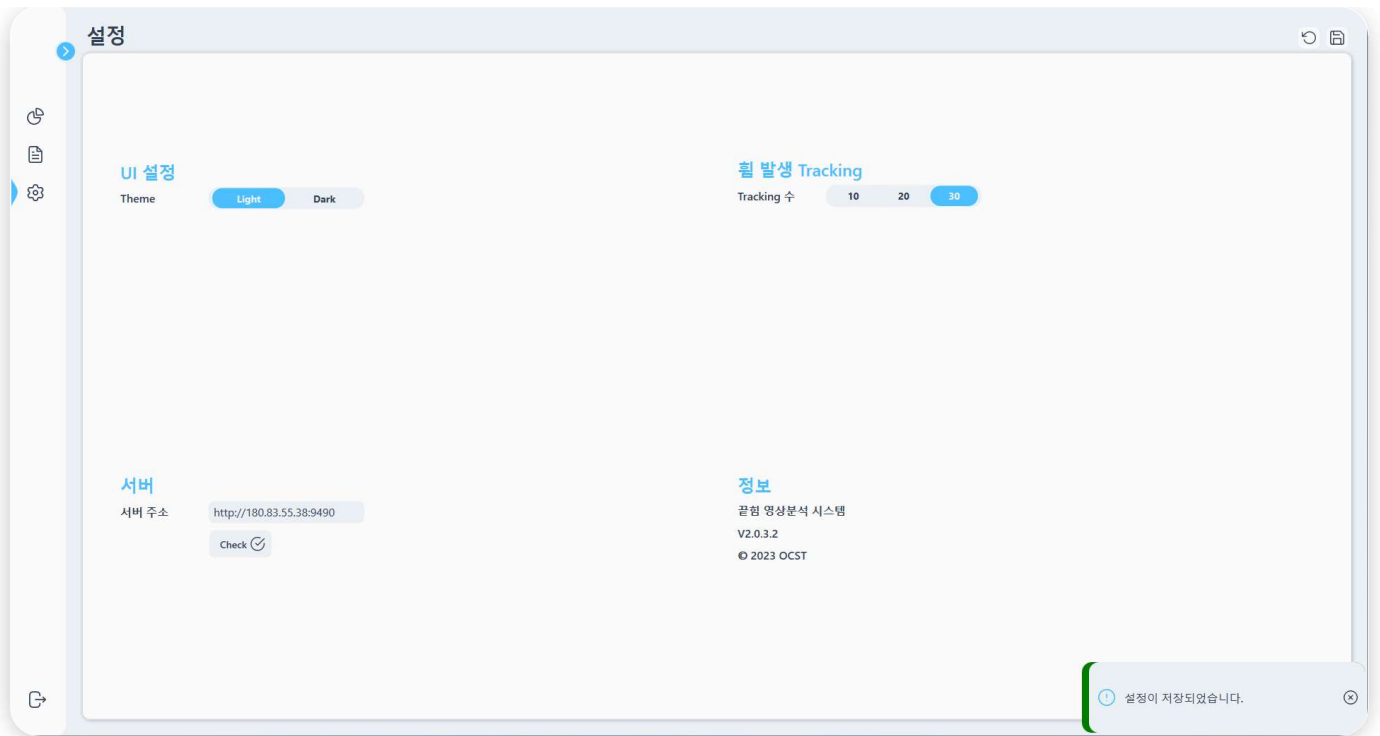
전체이력조회

1월 | 2024-06-03 | 2024-06-04 | OS2,OS1,0,DS1,DS2 | 검색 | 내보내기

OS2
 OS1
 0
 DS1
 DS2

날짜	추출시간	Heat No.	절단SEQ	휨각도	발생길이	휨수치	휨방향
2024-06-04	09:44:23	20240604094421	1	-0.96	16050.06	57.10	DS1
2024-06-04	09:43:57	20240604094355	1	2.06	15900.07	112.33	OS2
2024-06-04	09:43:52	20240604094350	1	-1.67	16050.63	100.53	DS2
2024-06-04	09:43:25	20240604094323	1	-0.63	15900.07	34.47	DS1
2024-06-04	09:43:20	20240604094318	1	0.57	14000.05	28.13	OS1
2024-06-04	09:42:53	20240604094251	1	-0.55	15950.06	31.78	DS1
2024-06-04	09:42:48	20240604094246	1	-1.99	15950.87	118.41	DS2
2024-06-04	09:42:21	20240604094219	1	1.42	14400.32	70.93	OS1
2024-06-04	09:42:16	20240604094213	1	0.94	12950.12	42.10	OS1
2024-06-04	09:41:49	20240604094147	1	0.03	0.00	1.91	0
2024-06-04	09:41:43	20240604094141	1	0.18	0.00	8.87	0
2024-06-04	09:41:17	20240604094115	1	0.33	0.00	12.97	0
2024-06-04	09:41:11	20240604094109	1	0.48	0.00	22.38	0
2024-06-04	09:40:45	20240604094043	1	0.60	13100.06	28.57	OS1
2024-06-04	09:40:40	20240604094038	1	-0.96	16050.20	57.10	DS1
2024-06-04	09:40:13	20240604094011	1	2.50	12651.65	133.84	OS2
2024-06-04	09:40:08	20240604094005	1	-2.12	16851.18	138.65	DS2
2024-06-04	09:39:41	20240604093939	1	-0.63	15900.07	34.47	DS1
2024-06-04	09:39:36	20240604093933	1	0.57	14000.05	28.13	OS1
2024-06-04	09:39:09	20240604093907	1	-0.96	17050.23	62.23	DS1
2024-06-04	09:39:04	20240604093902	1	0.82	14950.12	43.79	OS1
2024-06-04	09:38:37	20240604093835	1	1.42	14400.32	70.93	OS1

Total: 7586 | 100 | Page: 76 | Goto



§ Technologies Used

- **Frontend Development:** C# WPF (Windows Presentation Foundation)
- **Image Processing:** OpenCV, Object Tracking, Segmentation, Anomaly Detection
- **Code Structure:** MVVM (Model-View-ViewModel), Dependency Injection
- **Data Management:** Entity Framework, MSSQL Database
- **Hardware Integration:** PLC Communication

§ Conclusion

This project showcases the integration of advanced video analysis techniques with industrial automation to enhance quality control and operational efficiency. The system's robust architecture, user-friendly interface, and real-time processing capabilities ensure reliable and efficient monitoring of the end bending process in large rolling operations. My contributions span from interface design and backend development to database integration and PLC communication, demonstrating a comprehensive approach to developing sophisticated industrial systems.



PRODUCT INSPECTION SYSTEM

2024-03-26

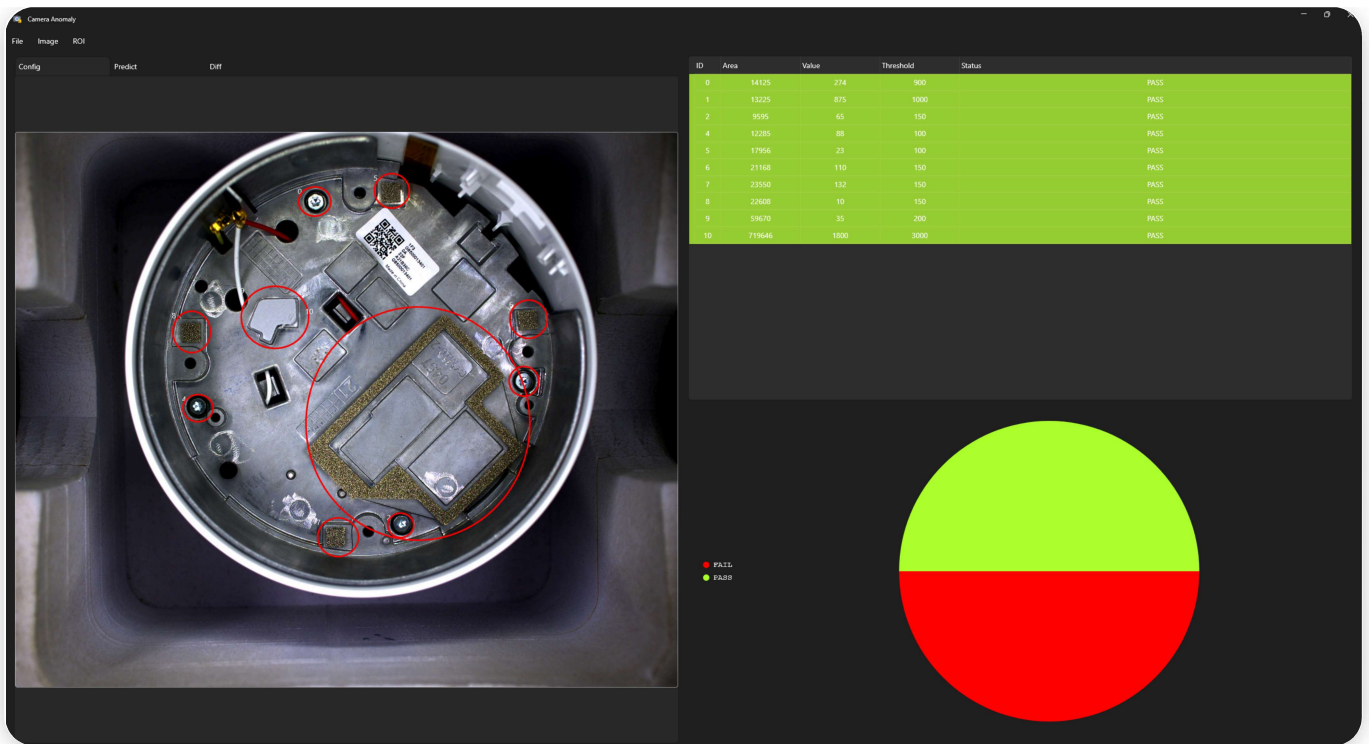
§ Project Summary

This project develops a vision-based system for automated product inspection on a conveyor belt. The system utilizes a camera to capture images of products as they move along the conveyor. These images are then compared to user-defined configuration images.

§ My Contribution

I implemented anomaly detection functionality, allowing users to define and track defect areas directly on reference images. Additionally, I integrated Entity Framework and an MSSQL database to save inspection result and config data.

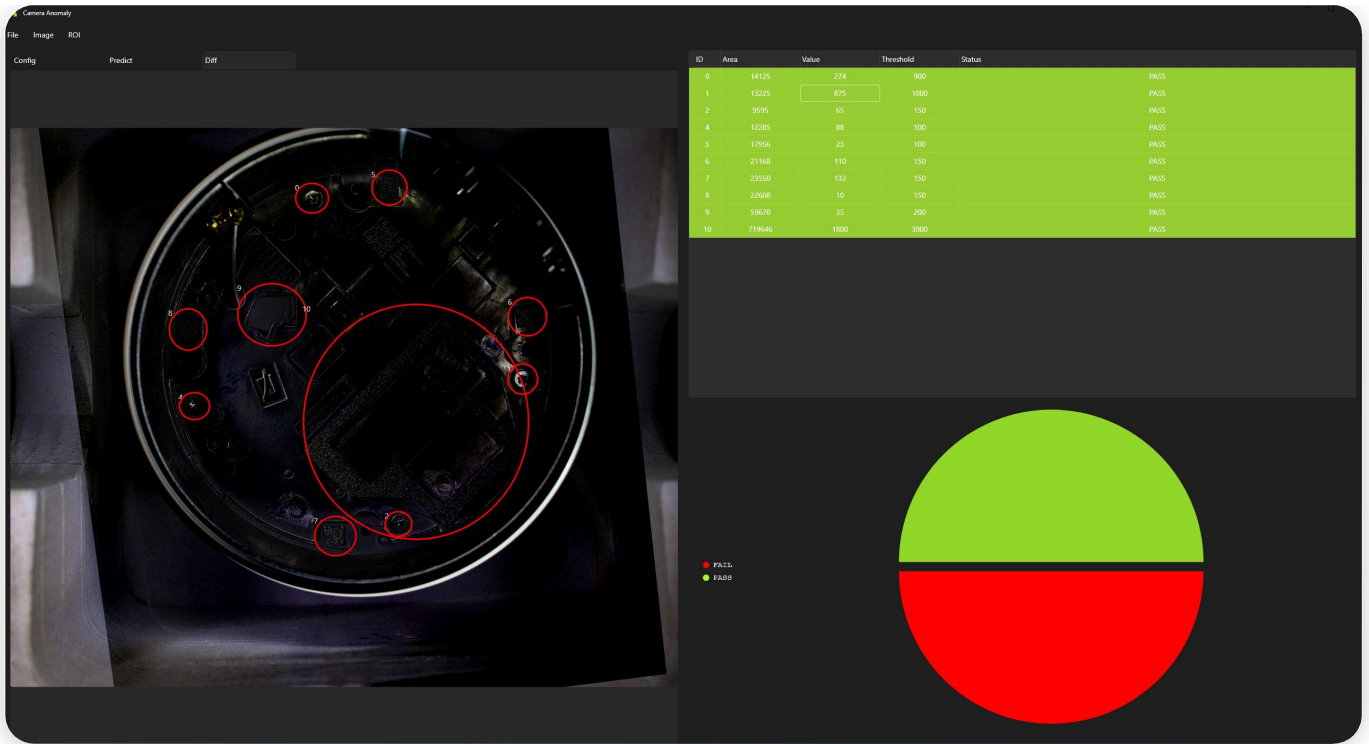




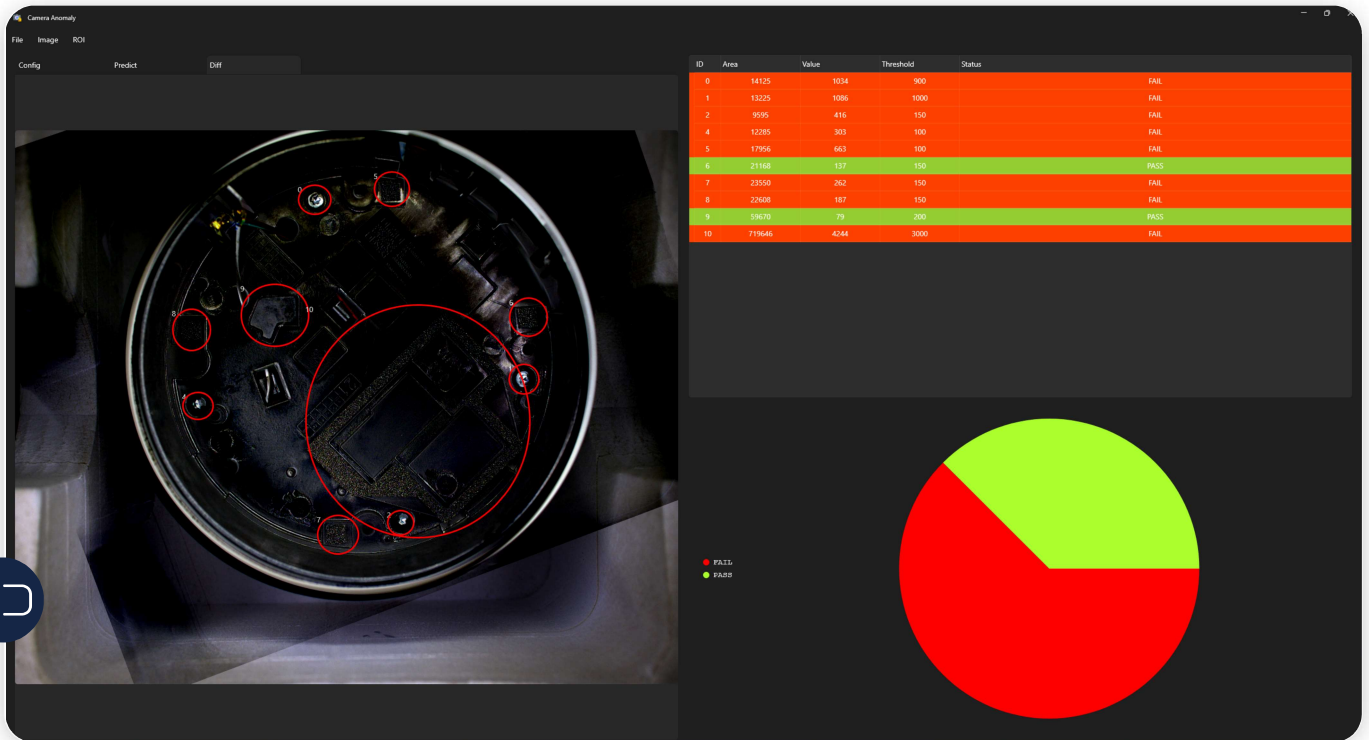
Anomaly region definitions

§ Key Features:

- **User-defined Anomaly Detection:** Users can mark specific regions directly on config images. The system identifies and flags products with anomalies in those designated areas, ensuring focus on critical inspection points.
- **OpenCV for Image Processing:** Leverages the powerful OpenCV library for various image processing tasks. This includes image capture from the camera, image comparison with reference images, and anomaly detection based on user-defined regions.
- **MVVM Design Pattern:** Employs the **MVVM** (Model-View-ViewModel) design pattern for a clean separation of concerns. This promotes maintainable and testable code, making future modifications and updates easier.
- **Database Integration:** The system integrates with **Entity Framework** and an MSSQL database for managing configuration data, such as reference images and anomaly region definitions.
- **PLC Communication:** The system communicates with a Programmable Logic Controller (PLC) to automate conveyor control based on inspection outcomes.



Inspection result with **PASS** result



Inspection result with **FAIL** result



Technologies:

- WPF (Frontend Development)
- OpenCV (Image Processing)
- MVVM, Dependence Injection (Code Structure)

- Entity Framework & MSSQL Database (Data Management)
- PLC Communication (Hardware Integration)

DOOR SAFETY SOLUTION

2023-09-01

§ Project Summary

This project, using *C#* and PostgreSQL in the WPF framework, is focused on safety in the factory. It lets maintenance engineers communicate with managers before entering machine areas. To access these areas, maintenance engineers need to call the manager for door opening permissions, ensuring safety.

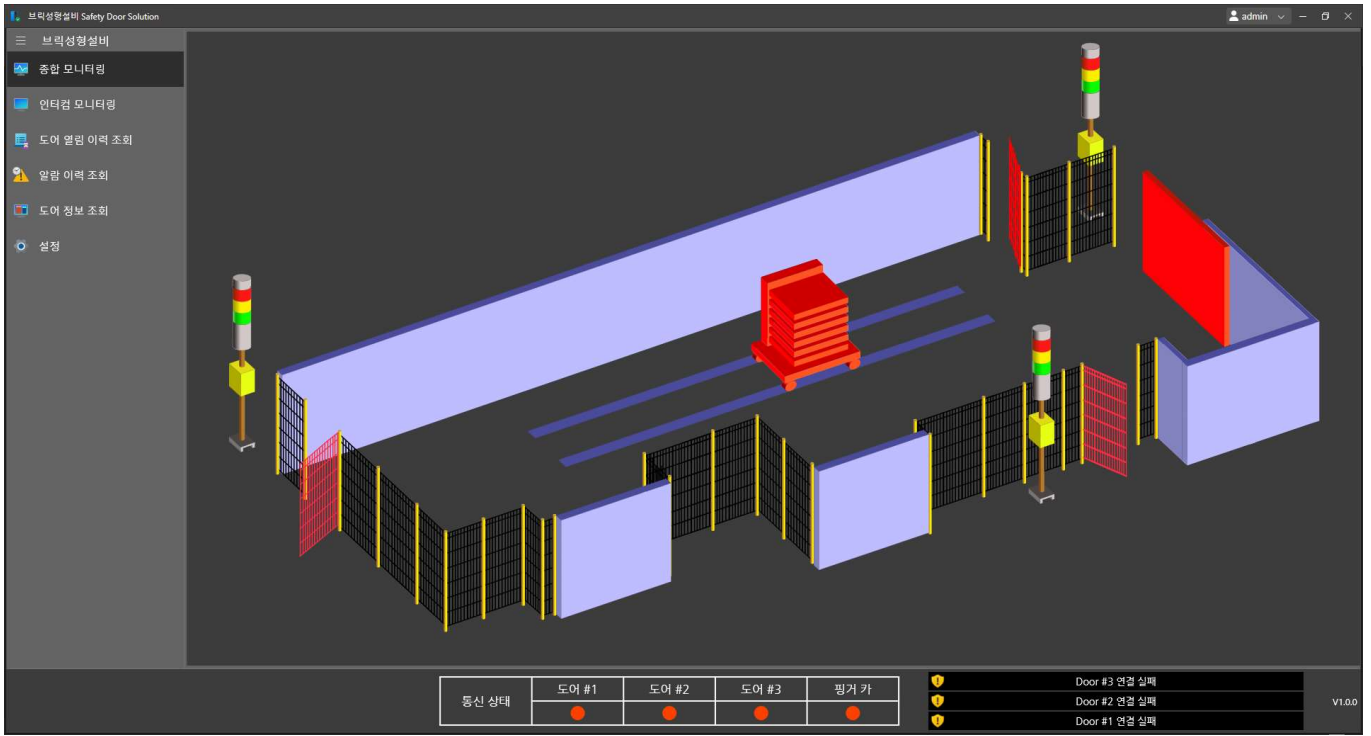
§ My Contribution

I developed the software and communication parts of the Door Safety Solution. There are two main parts in communication, PLC and intercom device. The software developed using WPF framework with MVVM, Dependency Injection design pattern; PostgreSQL and Entity Framework as database; communicate with PLC and intercom by TCP/IP protocol.

§ Factory Monitoring

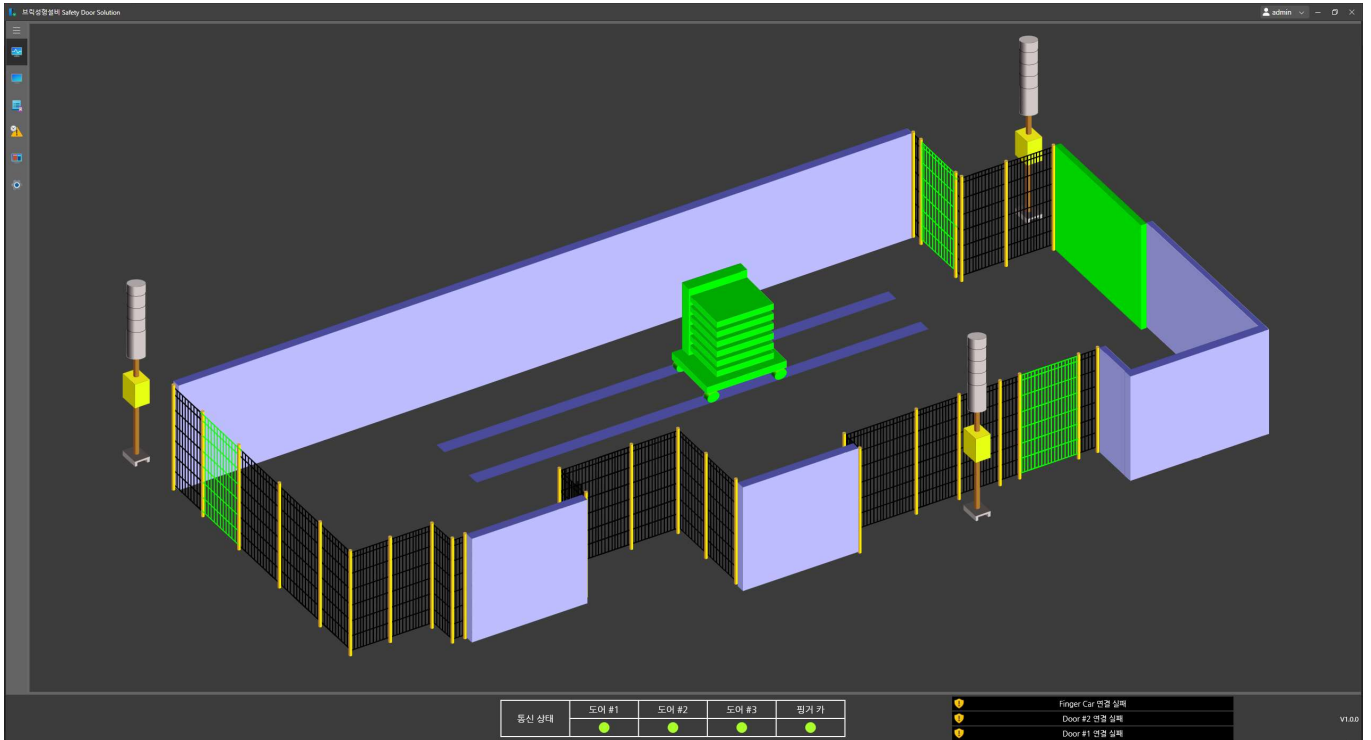
Shows a wide view of the factory with expanded menu.





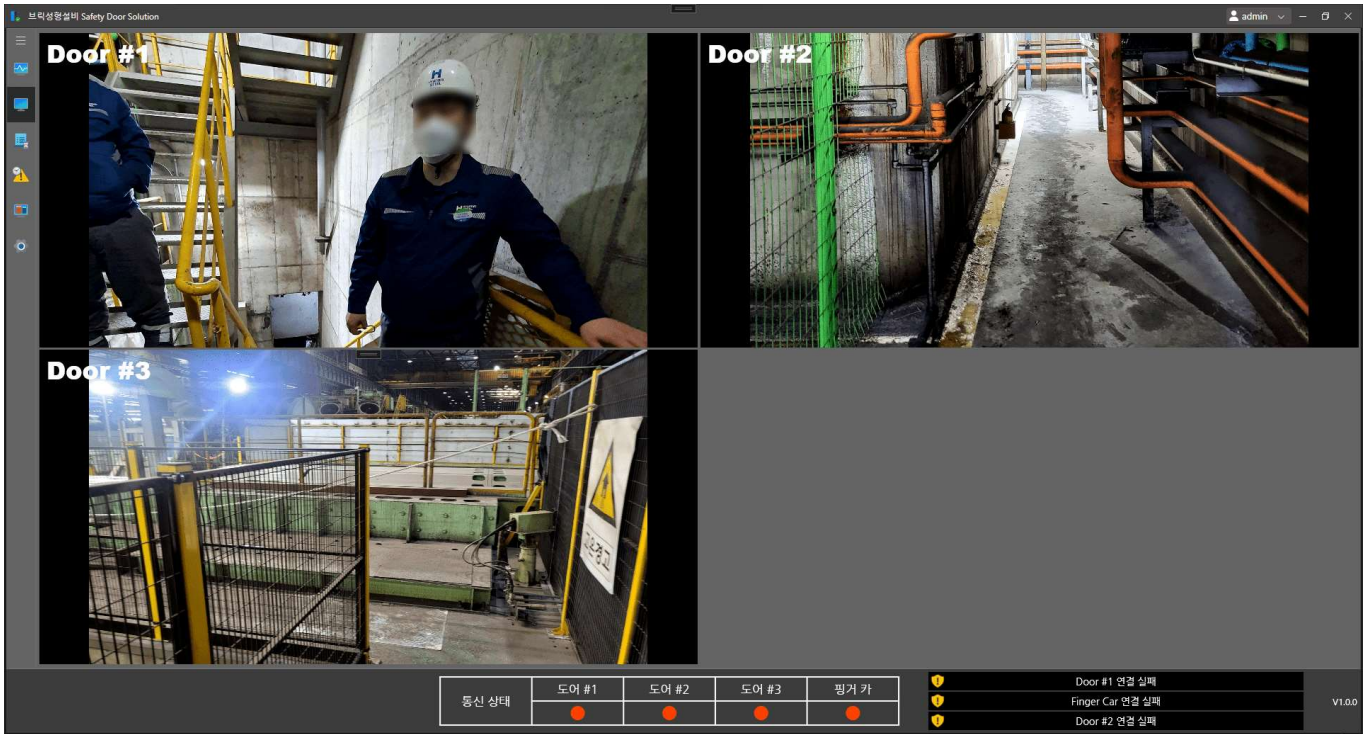
§ Compact Factory Monitoring

The menu can be displayed in compact mode to maximize view area.



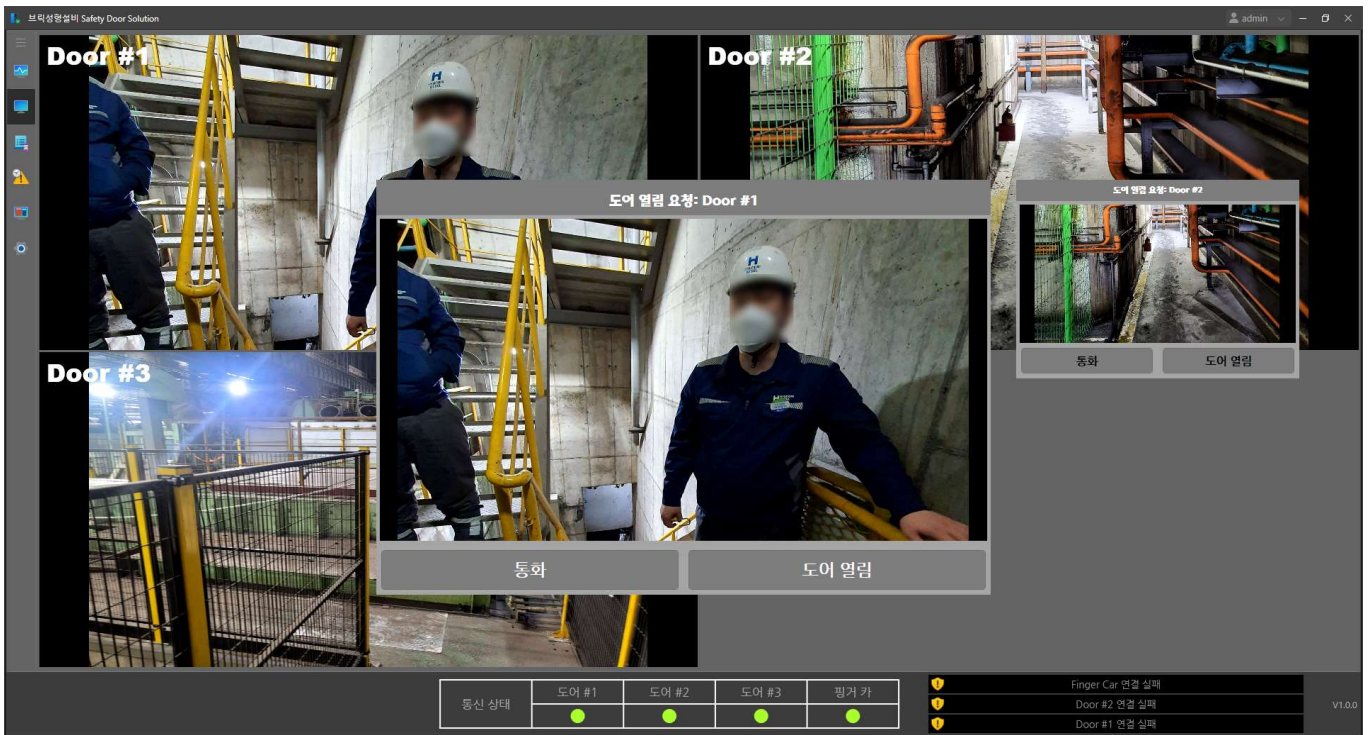
§ Video Monitoring

The layout automatically changes depending on the number of intercom device.



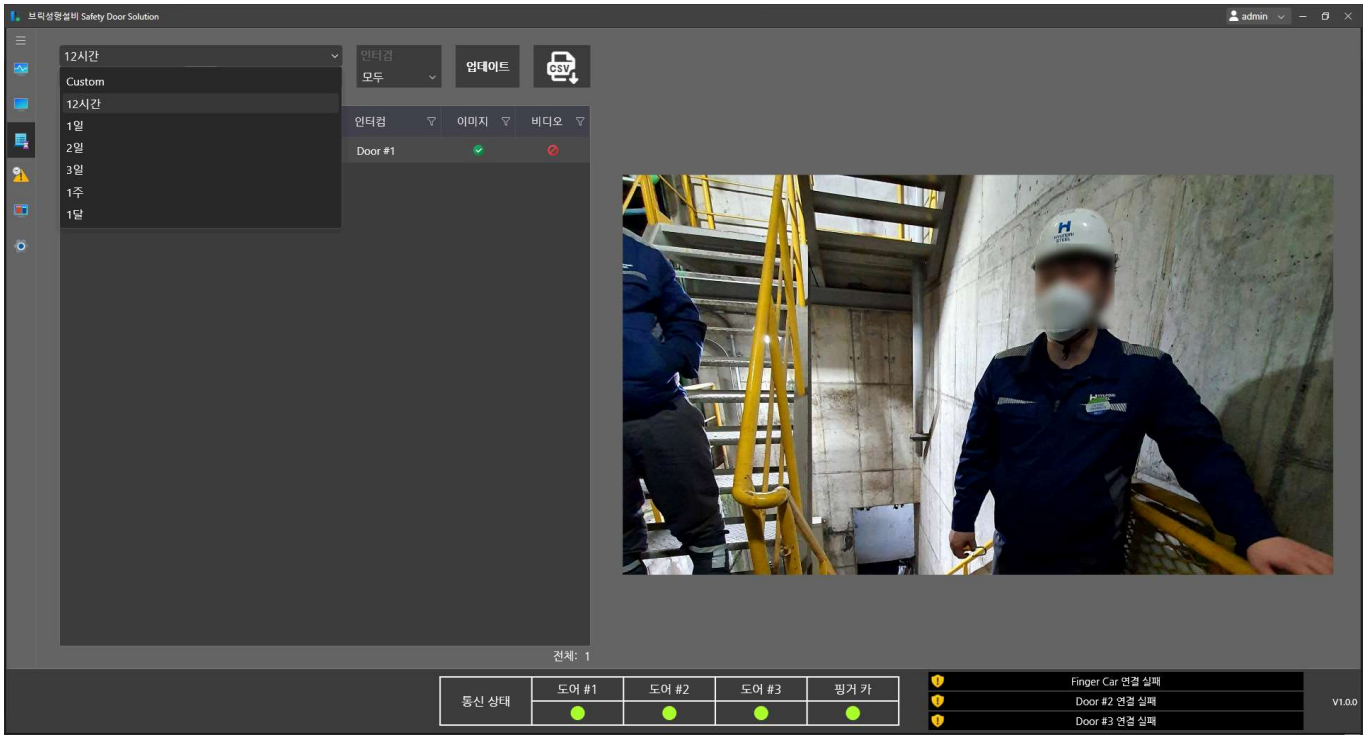
§ Popup Video Call

The popup is displayed when maintenance engineer make a call at access door. Another incoming call will be in wait list and stack at the right of program.



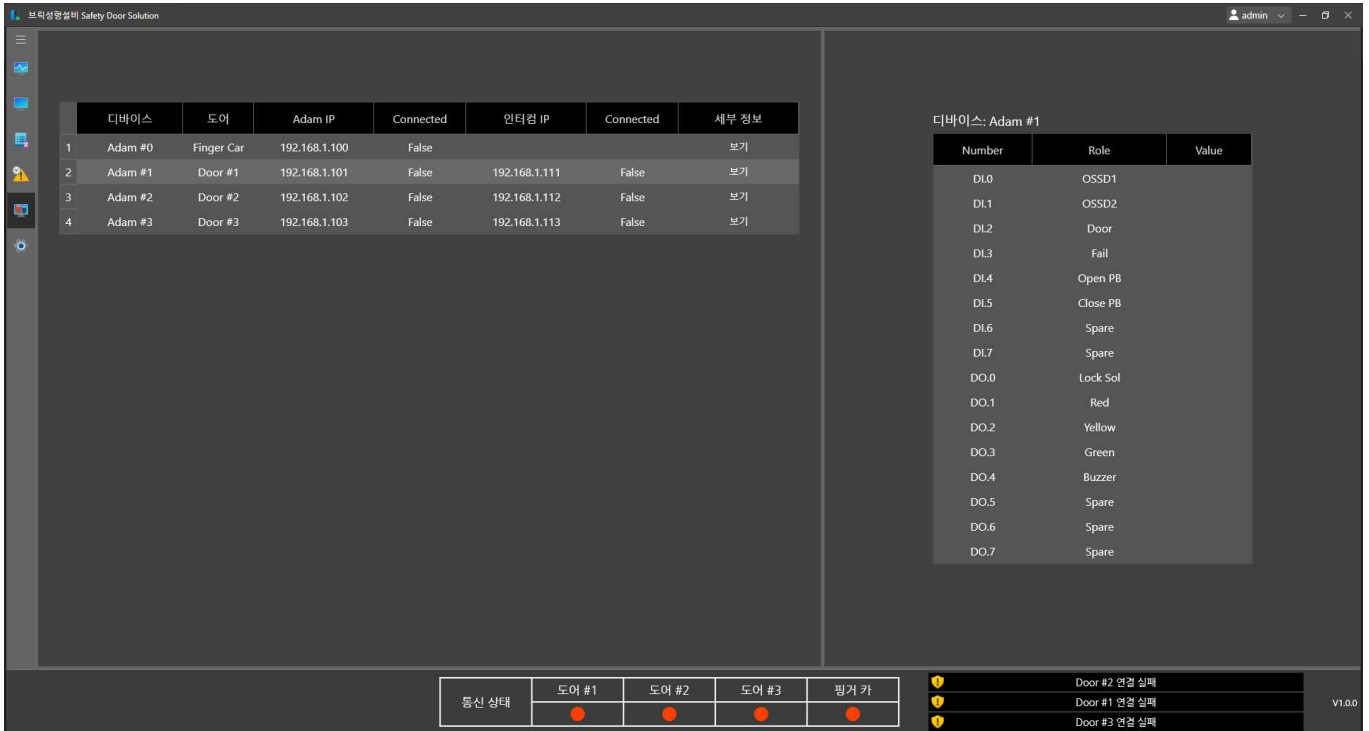
§ History Checking

People in the control room can look back at who used each door, with pictures and videos.



§ Device Monitoring

Shows data from machines in real-time.



§ Settings

Lets users change settings based on their role. Admins can decide which doors can be used, set maintenance mode to open all door at the same time, ...

도어 사용 설정

도어 사용 설정

- Door #1 사용
- Door #2 사용
- Door #3 사용

비밀번호 확인

저장 취소

통신 상태	도어 #1	도어 #2	도어 #3	핑거 카
	●	●	●	●

- Door #3 연결 실패
- Door #1 연결 실패
- Finger Car 연결 실패

admin

VT1.0.0



 thanhdat6716

TC-NET DATA SERVER

2023-06-01

§ Project Summary

TC-net is a network technology developed by Toshiba that is specifically designed for industrial automation applications that require high reliability and high-speed communication. TC-net Data Server is a Windows application developed in C# and PostgreSQL. It provides real-time data gathering in high-speed and high-volume data, and stores data in a time series database for further analysis and processing. Multithreading and asynchronous programming are used to ensure high performance and reliability.

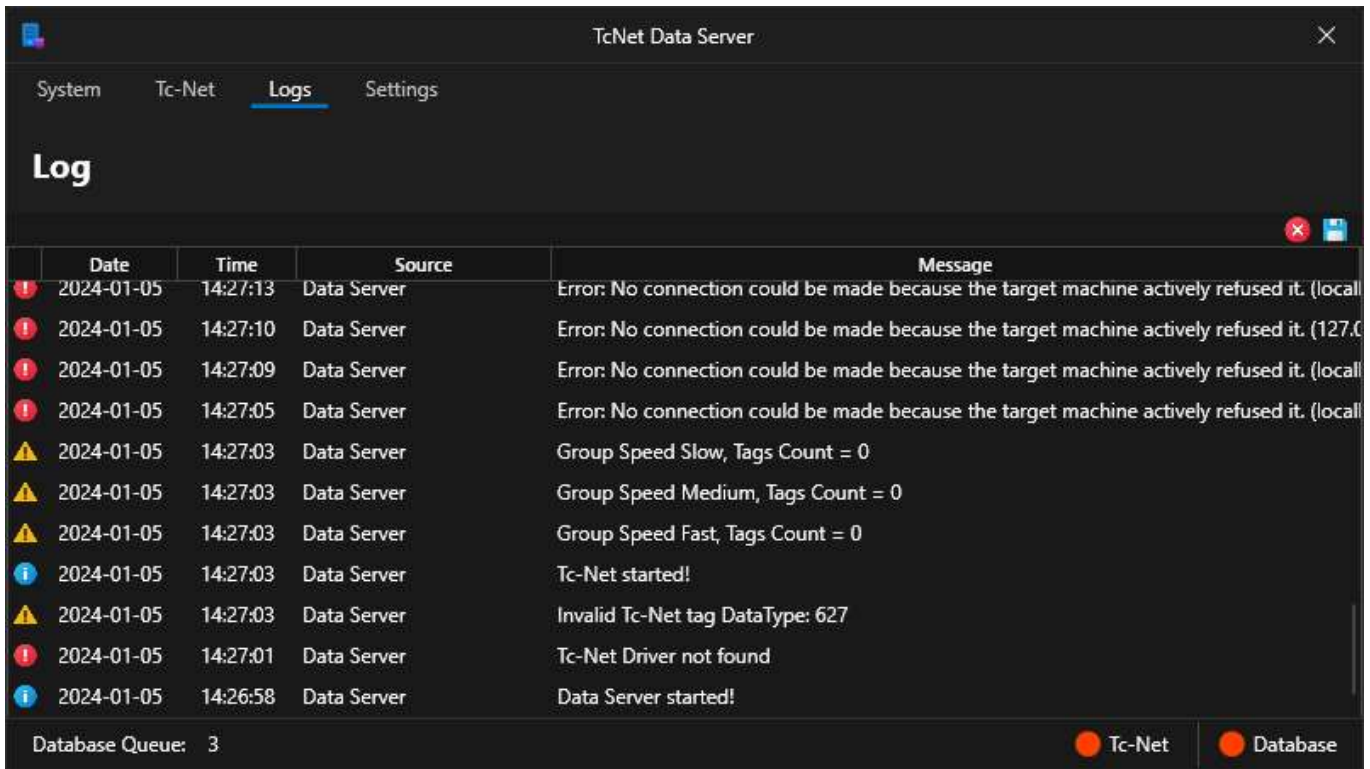
§ My Contribution

I was responsible for the development and maintenance of the TC-net Data Server, including front-end and back-end development using C#. I also integrated the system with PostgreSQL to store data item information and VictoriaMetrics (time series database) for effective data management and analysis.

§ TC-net Data Server Log

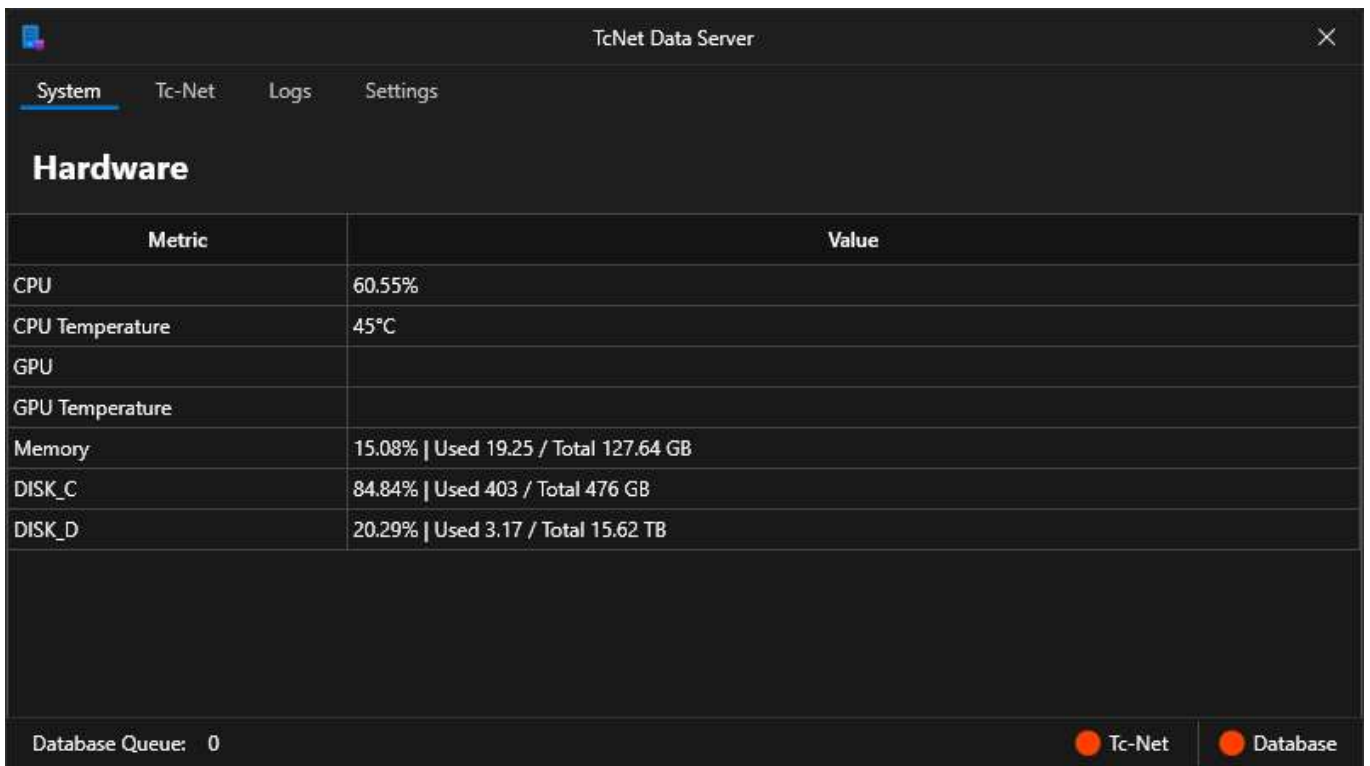
Displays the log of the TC-net Data Server, including information on data item values, errors, and more.





§ Server hardware information

Displays information on the server hardware, including CPU, memory, and disk usage.



§ Data Item View

Displays capture speed of data items.

The screenshot shows the 'TcNet Data Server' application window with the 'TC-NET' tab selected. The interface includes a navigation bar with 'System', 'Tc-Net', 'Logs', and 'Settings'. Below the navigation bar is a table with the following data:

Tag Speed	Tag Count	Check (ms)	Actual (ms)	Minimum (ms)	Maximum (ms)
Fast	0	10	0.00	0.00	0.00
Medium	0	50	0.00	0.00	0.00
Slow	0	500	0.00	0.00	0.00
VerySlow	7968	1000	0.00	0.00	0.02
Qualifying	61560	30000	0.00	0.00	0.01

At the bottom of the window, the 'Database Queue' is shown as 2. There are two status indicators: 'Tc-Net' and 'Database', both represented by red circles.

§ Server and database configuration

To handle high-speed and high-volume data, settings can be configured to change database storage.

The screenshot shows the 'TcNet Data Server' application window with the 'Settings' tab selected. The interface includes a navigation bar with 'System', 'Tc-Net', 'Logs', and 'Settings'. Below the navigation bar is the 'Settings' section with the following configuration fields:

- Communication Port: 54321
- VMs Uri: http://127.0.0.1:40001
- VMs Max Queue: 10000
- VMs batch size: 10

At the bottom of the window, the 'Database Queue' is shown as 0. There are two buttons: 'Save' and 'Restore'. There are also two status indicators: 'Tc-Net' and 'Database', both represented by red circles.



 New Possibilities

 thanhdat6716@gmail.com

 +82-10-4409-6699

 thanhdat6716

WAREHOUSE MANAGEMENT SYSTEM

2023-03-01

§ Project Summary

This project is a Windows application developed in C#, Python, and SQL Server. It provides real-time monitoring and tracking for efficient management of customer goods in warehouses. The system also enhances warehouse safety with CCTV monitoring and object detection features.

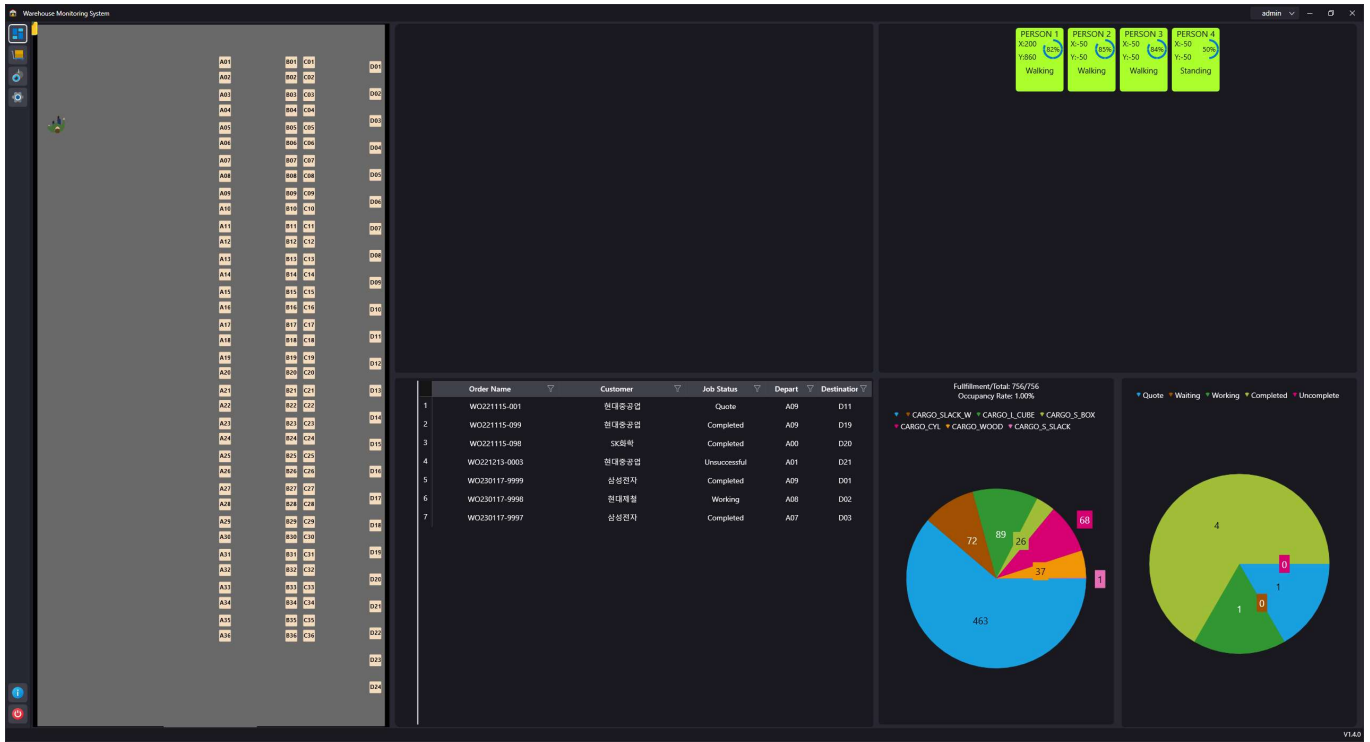
§ My Contribution

I was responsible for the development and maintenance of the software components of the Warehouse Management System, including front-end and back-end development using C# and Python. I also integrated the system with SQL Server for effective data management.

§ Warehouse Management System (WMS) Dashboard

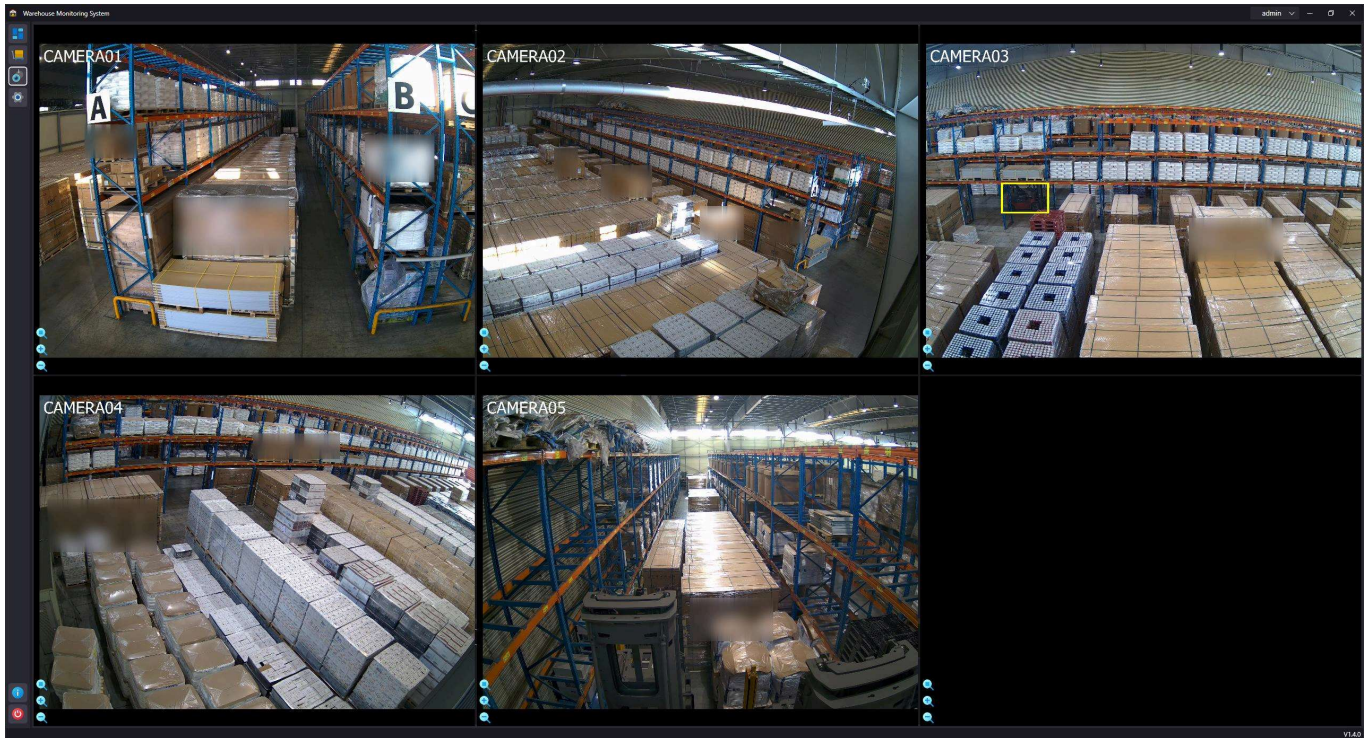
The main dashboard and homepage of the system, tracking the positions of forklifts and person in the warehouse and providing stock status updates.





§ WMS Camera

Displays live camera feeds and surveillance in the warehouse, utilizing an object detection algorithm to identify and track objects.



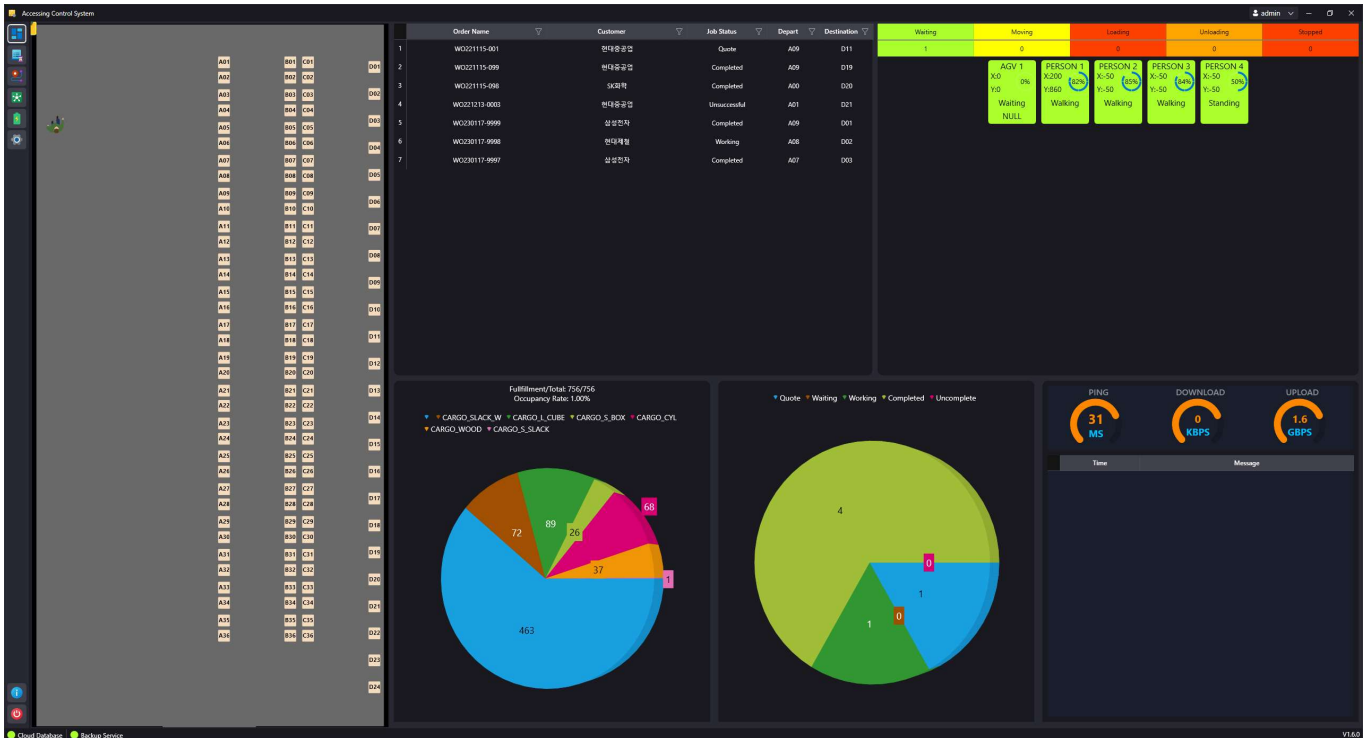
§ User Management

Handles administration and configuration of user access to the WMS.



§ Access Control System (ACS) Dashboard

The ACS dashboard shows the status of IoT devices, including position, working status, tasks, battery life, and more.



§ ACS Order

Manages and processes orders within the ACS.

Order Name	Customer	Job Status	Quantity	Depart	Destination	Cargo Tag	Order Time	Finish Time	Expired Time
WO221115-001	현대중공업	Queue	1	A09	D11	M2211-2020-0170			
WO221115-099	현대중공업	Completed	99	A09	D19	M2211-2020-0170	2023-06-30 16:19:09		
WO221115-098	SK화학	Completed	98	A00	D20	M2211-2020-0170	2023-06-30 16:19:09		
WO221213-0003	현대중공업	Unsuccessful	3	A01	D21	M2211-2020-0170	2023-06-30 16:19:09		
WO230117-9999	삼성전자	Completed	99	A09	D01	M2211-2020-0170	2023-06-30 16:19:09		
WO230117-9998	현대제철	Working	98	A08	D02	M2211-2020-0170	2023-06-30 16:19:09		
WO230117-9997	삼성전자	Completed	97	A07	D03	M2211-2020-0170	2023-06-30 16:19:09		

§ ACS Device Status

Monitors the status and connectivity of ACS devices.

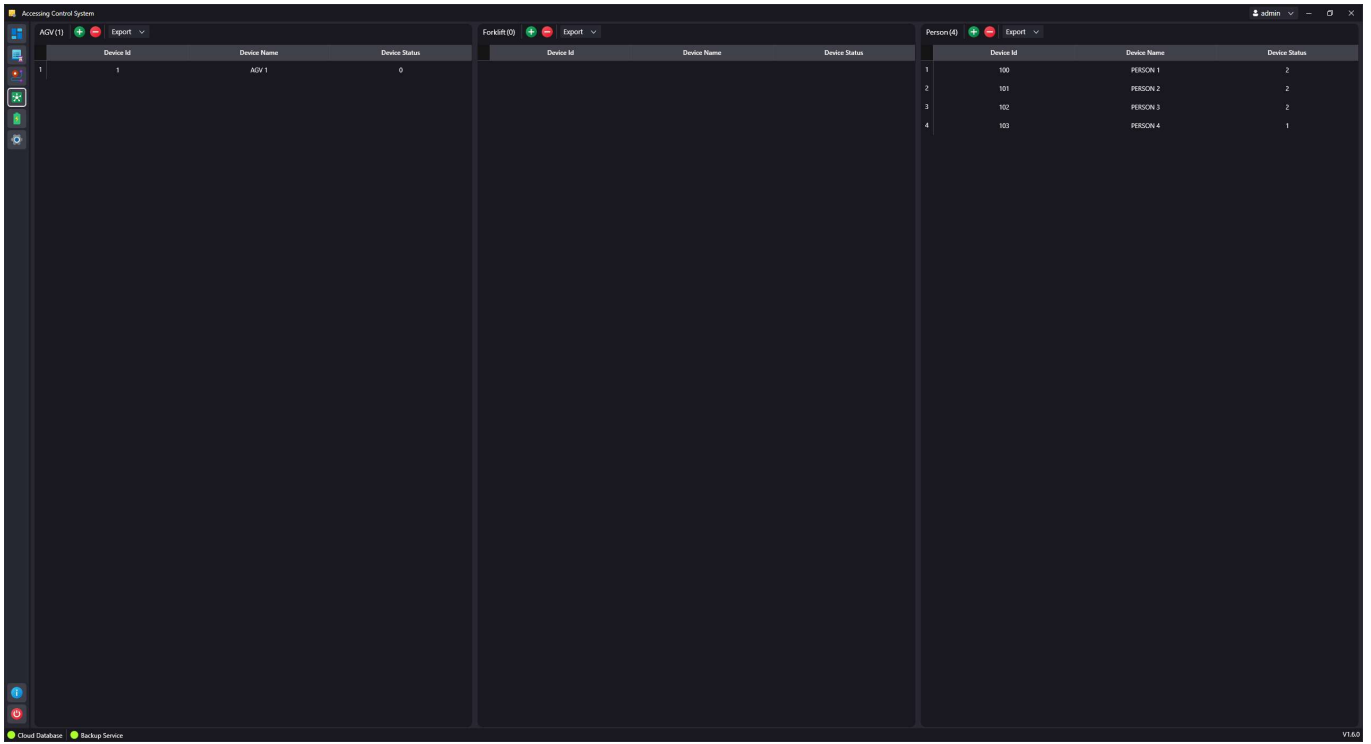
디바이스	도어	Adam IP	Connected	인터컴 IP	Connected	세부 정보
Adam #0	Finger Car	192.168.1.100	False			보기
Adam #1	Door #1	192.168.1.101	False	192.168.1.111	False	보기
Adam #2	Door #2	192.168.1.102	False	192.168.1.112	False	보기
Adam #3	Door #3	192.168.1.103	False	192.168.1.113	False	보기

Number	Role	Value
DI.0	OSSD1	
DI.1	OSSD2	
DI.2	Door	
DI.3	Fail	
DI.4	Open PB	
DI.5	Close PB	
DI.6	Spare	
DI.7	Spare	
DO.0	Lock Sol	
DO.1	Red	
DO.2	Yellow	
DO.3	Green	
DO.4	Buzzer	
DO.5	Spare	
DO.6	Spare	
DO.7	Spare	

통신 상태	도어 #1	도어 #2	도어 #3	핑거 카	Door #2 연결 실패	Door #1 연결 실패	Door #3 연결 실패
	●	●	●	●			

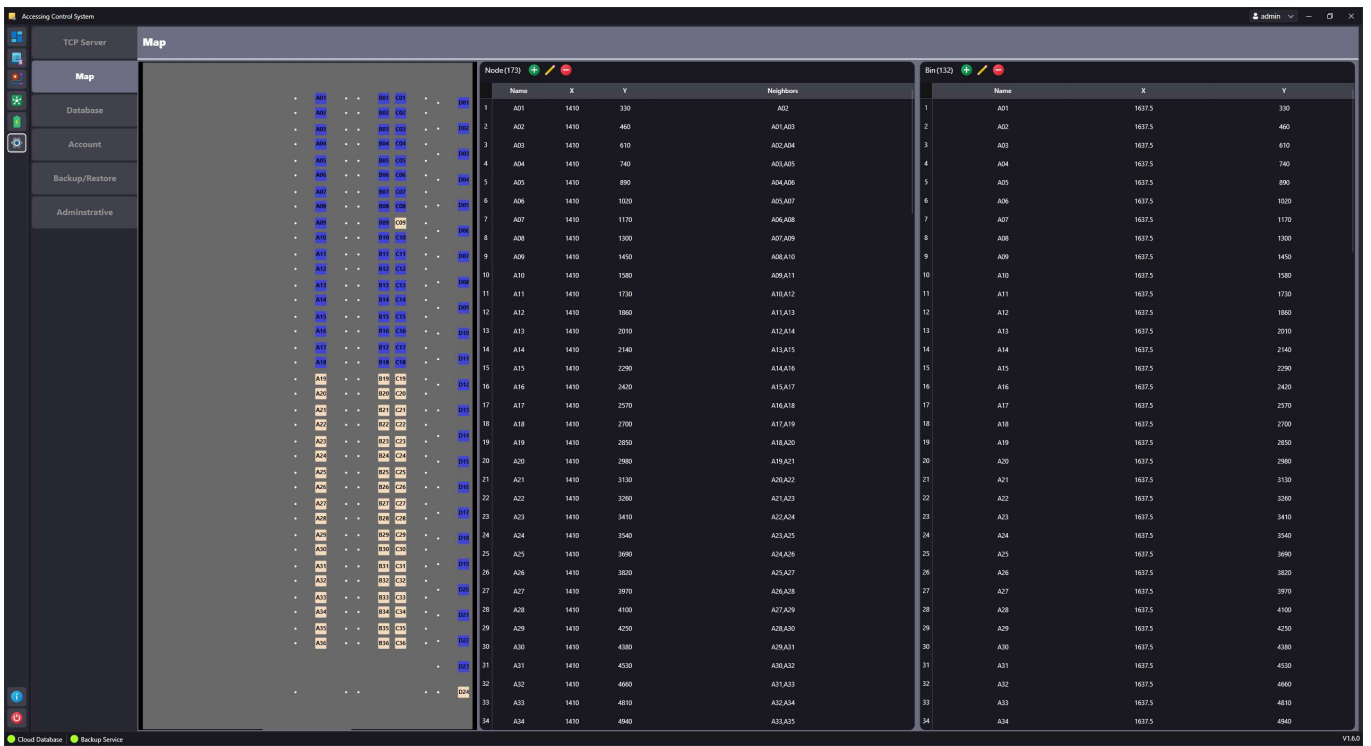
§ ACS Device Management

Allows for the management and real-time monitoring of ACS devices.



§ ACS Layout

Enables configuration of the warehouse layout, including rack and road placement.



§ ACS Routing

Provides an A* algorithm for finding optimal routes for AGV devices.



✉ thanhdat6716@gmail.com

☎ +82-10-4409-6699

in thanhdat6716

VIDEO SAFETY SOLUTION

2022-09-01

§ Project Summary

This project utilizes AI-based video analysis to detect cobbles and people in danger zones. It monitors all CCTV cameras in real-time, allowing for the setting and storing of danger zones. The system issues real-time alarms during danger situations and communicates with the video surveillance PC and factory PLC. The PLC and network are also integrated to control equipment operations in case of cobble occurrences.

§ My Contribution

I contributed to the development and maintenance of the Video Safety Solution. My tasks included AI-based video anomaly detection, real-time monitoring, and integrating PostgreSQL and Python for data processing and analysis.

§ Video Monitoring

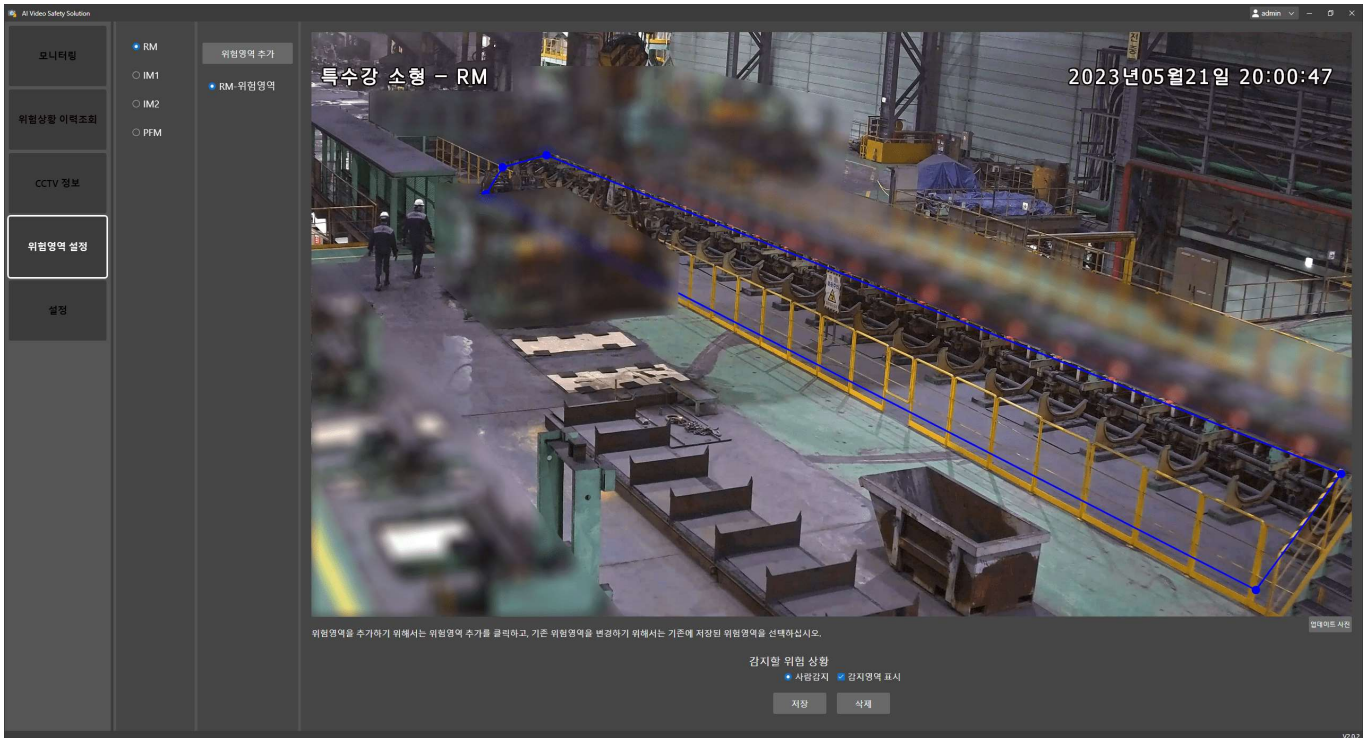
Facilitates real-time monitoring of live video streams from surveillance cameras. Object detection algorithms are used to identify and track person in the video streams.





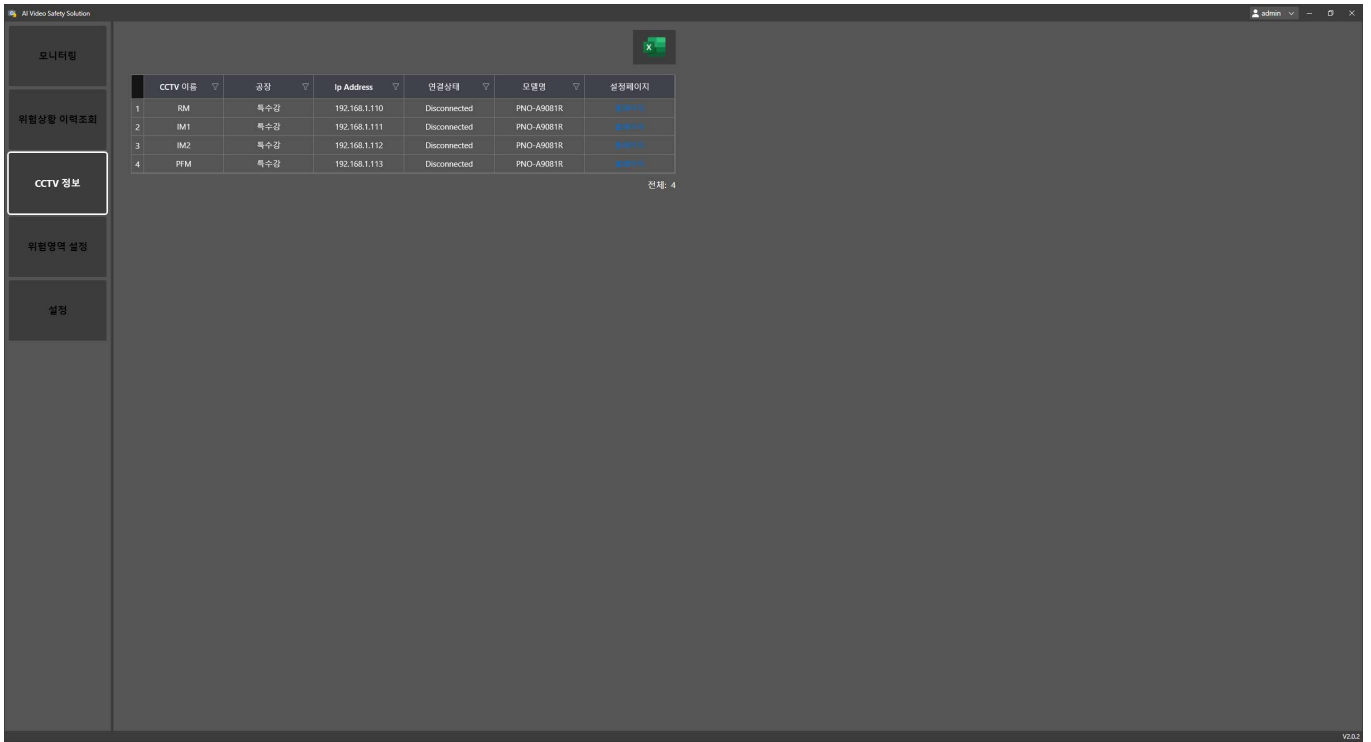
§ Area Settings

Allows for the configuration of specific dangerous areas. It triggers alarms when people enter these areas while machinery is operating.



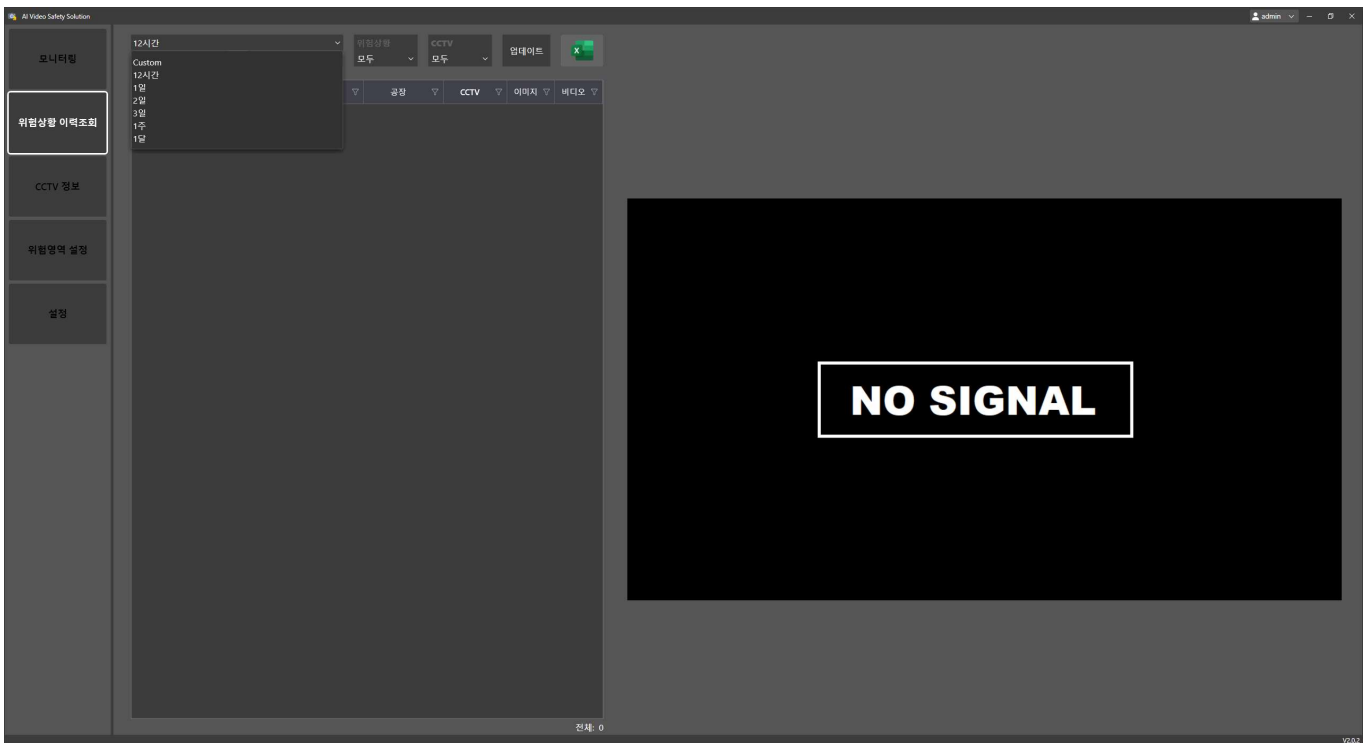
§ Device Information

Provides detailed information on video monitoring devices.



§ History

Allows viewing of historical data and events related to video monitoring.



§ Video Safety Settings

Manage settings and configurations of video safety components.



New Thinking
New Possibilities

 thanhdat6716@gmail.com

 +82-10-4409-6699

 thanhdat6716

DATA CAPTURE SYSTEM

2022-06-01

§ Project Summary

This project involved developing a WPF application for reading large amount of tag data from OPC UA and writing to InfluxDB (time series database), and a TypeScript Grafana extension for real-time and historical data monitoring.

§ My Contribution

I developed the WPF application and the Grafana extension, focusing on front-end development and InfluxDB integration.

§ Data Item View

Displays real-time data item values.



Log	Address	Type	Tag Name	Interval	Source Name	Data Type	Description	Initial Value	EngUnit	MinEU	MaxEU	MinRaw	MaxRaw
ms=3=1 OPC-UA	L2_ACT_FCE0INS_000	0	Source0	0	Source0	SPARE250248A		0	kg	2	2	2	2
ms=3=1 OPC-UA	L2_ACT_FCE0INS_001	0	Source0	0	Source0	DC AUX POWER TR FAULT C305-FAULT250248A		50.5	m/s	3	3	3	3
ms=3=1 OPC-UA	L2_ACT_FCE0INS_002	0	Source0	0	Source0	No.3 DESCALING PUMP 50/51 NF320-51250248B		90.5	kg	4	4	4	4
ms=3=1 OPC-UA	L2_ACT_FCE0INS_003	0	Source0	0	Source0	FCE ENT R/T E-STOPL1-L6) PEMG1005250249A		875	m/s	5	5	5	5
ms=3=1 OPC-UA	L2_ACT_FCE0INS_004	0	Source0	0	Source0	DC ROT E-STOP(K1-K8) PEMG5005250249B		7	kg	6	6	6	6
ms=2=5 OPC-UA	L2_ACT_FCE0INS_005	0	Source0	0	Source0	Inlet Air Temperature(degC)		50	kg	2	2	2	2
ms=2=5 OPC-UA	L2_ACT_FCE0INS_006	0	Source0	0	Source0	Winding-RDS) Temperature(degC)		1000	m/s	3	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_007	0	Source0	0	Source0	BEARING TEMP (C-LOAD) 0-100degC/4-20mA\$250274A		4	m/s	4	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_008	0	Source0	0	Source0	WINDING TEMP-3 0-200degC/4-20mA\$250274A		5	kg	5	4	4	4
ms=2=5 OPC-UA	L2_ACT_FCE0INS_009	0	Source0	0	Source0	INLET AIR (2) 0-100degC/4-20mA\$250274A		6	m/s	6	5	5	5
ms=2=5 OPC-UA	L2_ACT_FCE0INS_010	0	Source0	0	Source0	OUTLET AIR 0-100degC/4-20mA\$250274A		1	kg	1	6	6	6
ms=2=5 OPC-UA	L2_ACT_FCE0INS_011	0	Source0	0	Source0	WINDING TEMP-3 F3-T3 0-200degC/4-20mA\$250271A		2	kg	2	2	2	2
ms=2=5 OPC-UA	L2_ACT_FCE0INS_012	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_FCE3E_501		2	kg	2	2	2	2
ms=2=5 OPC-UA	L2_ACT_FCE0INS_013	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_FCE3E_502		0	m/s	3	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_014	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_FCE3E_502A		90.5	kg	4	4	4	4
ms=2=5 OPC-UA	L2_ACT_FCE0INS_015	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_SLBWGH_501		50.5	m/s	2	2	2	2
ms=2=5 OPC-UA	L2_ACT_FCE0INS_016	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_SLBWGH_502		875	kg	3	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_017	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_SLBWGH_503		7	kg	2	2	2	2
ms=2=5 OPC-UA	L2_ACT_FCE0INS_018	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_SLBWGH_506		50	m/s	3	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_019	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_SLBWSR_502		1000	m/s	4	4	4	4
ms=2=5 OPC-UA	L2_ACT_FCE0INS_020	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_SLBWSR_503		4	kg	5	5	5	5
ms=2=5 OPC-UA	L2_ACT_FCE0INS_021	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_ABTBL_LK_CND		5	m/s	6	6	6	6
ms=2=5 OPC-UA	L2_ACT_FCE0INS_022	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_ABTBL_LUK_CND		6	kg	2	2	2	2
ms=2=5 OPC-UA	L2_ACT_FCE0INS_023	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_FCE4E_501		1	kg	3	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_024	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_FCE4E_502		2	kg	4	3	3	3
ms=2=5 OPC-UA	L2_ACT_FCE0INS_025	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_FCE4E_502A		2	m/s	5	4	4	4
ms=2=5 OPC-UA	L2_ACT_FCE0INS_026	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_PHS190_SML		2	kg	6	5	5	5
ms=2=5 OPC-UA	L2_ACT_FCE0INS_027	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_PHS190_SML		2	m/s	1	6	6	6
ms=2=5 OPC-UA	L2_ACT_FCE0INS_028	0	Source0	0	Source0	TC-net 100LAN.M2000-STN1.I.O_PHS190_SML		2	kg	2	2	2	2

§ Data Item Configuration

Allows configuration of data item settings.

Tag Manager

Tag Name: L2_ACT_FCE0INS_003 OPC-UA 0

Comment: FCE ENT R/T E-STOPL1-L6) PEMG1005250249A

Source: Source0

Address: ns=3=1005

Server Tag Name: Use This Name as Tag Name

Log Data Retentive Value

Initial Value: 875 Min EU: 5 Max EU: 5

Eng Units: m/s Min Raw: 5 Max Raw: 5

OK Cancel

Grafana View

A Grafana extension for monitoring time series data in real-time and accessing historical data.



+82-10-4409-6699

in thanhdat6716



CCTV RECORDER

2021-09-01

§ Project Summary

CCTV Recorder, built using C# on WPF, collects factory video for AI training. It features video controls like Zoom, Rotate, and Record, and allows CCTV configuration for recording purposes.

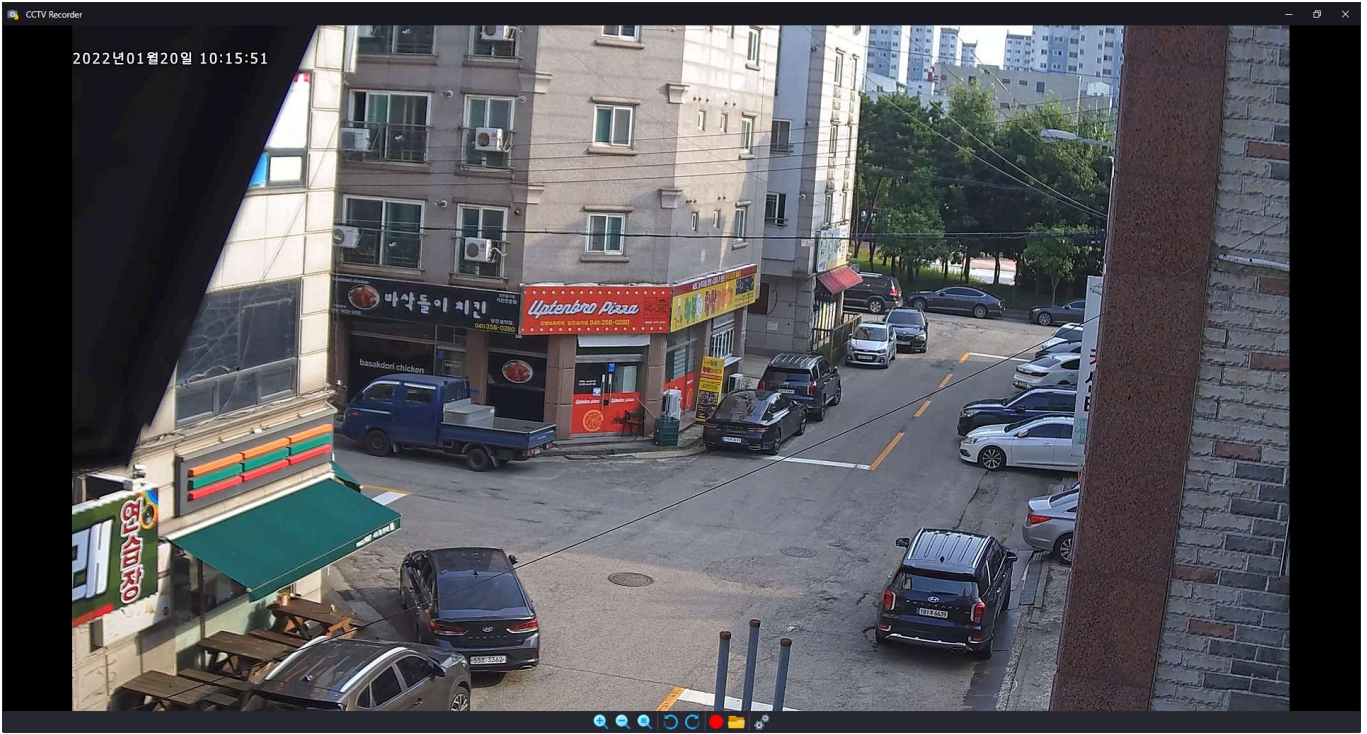
§ My Contribution

I contributed to both front-end and back-end development, integrating Python for AI training and video processing.

§ Monitoring CCTV

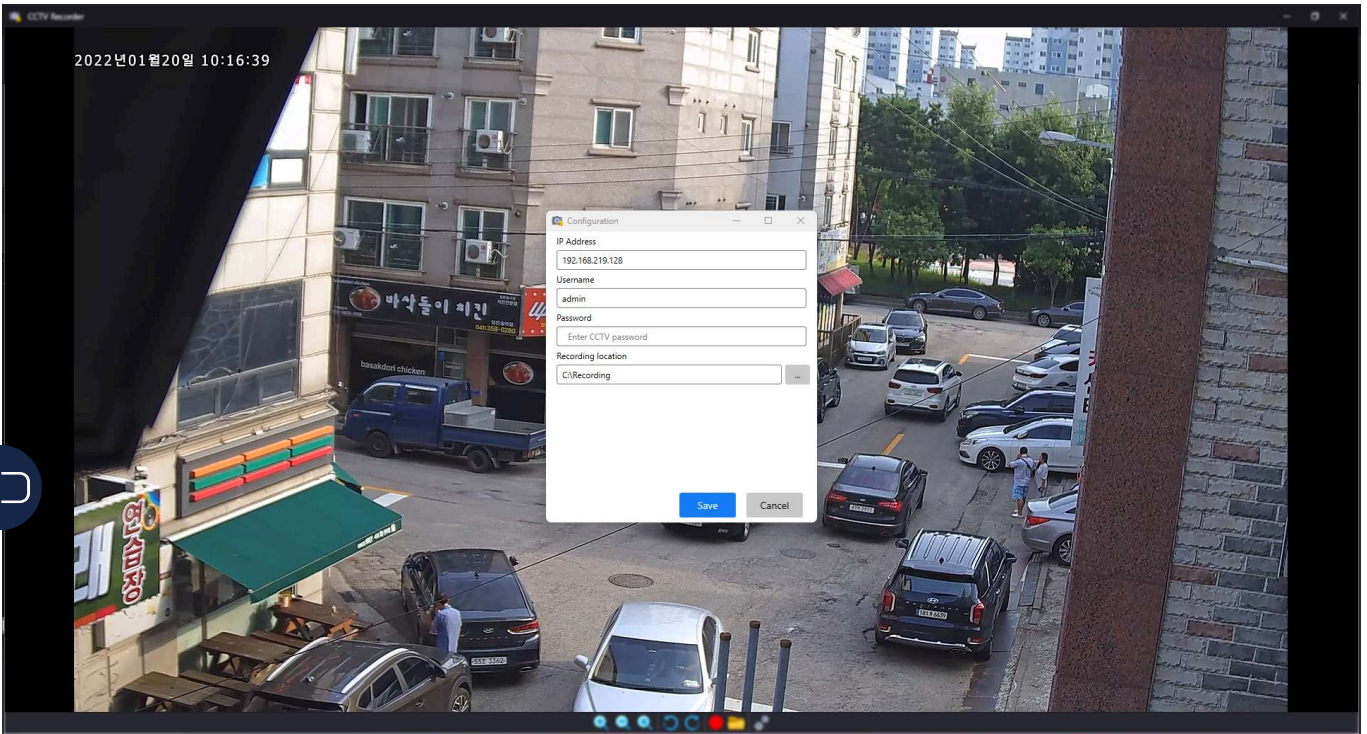
Enables real-time video stream monitoring.





§ Recording and Settings

Configures camera recording settings.



+82-10-4409-6699

in thanhdat6716

REBAR TRACKING

2021-06-01

§ Project Summary

Rebar Tracking is a Python application with tkinter as its GUI framework, using PostgreSQL for database management. This project designed to monitor and track rebar movement direction and count in a steel factory, offering real-time updates and capturing the transportation path of rebar within the facility.

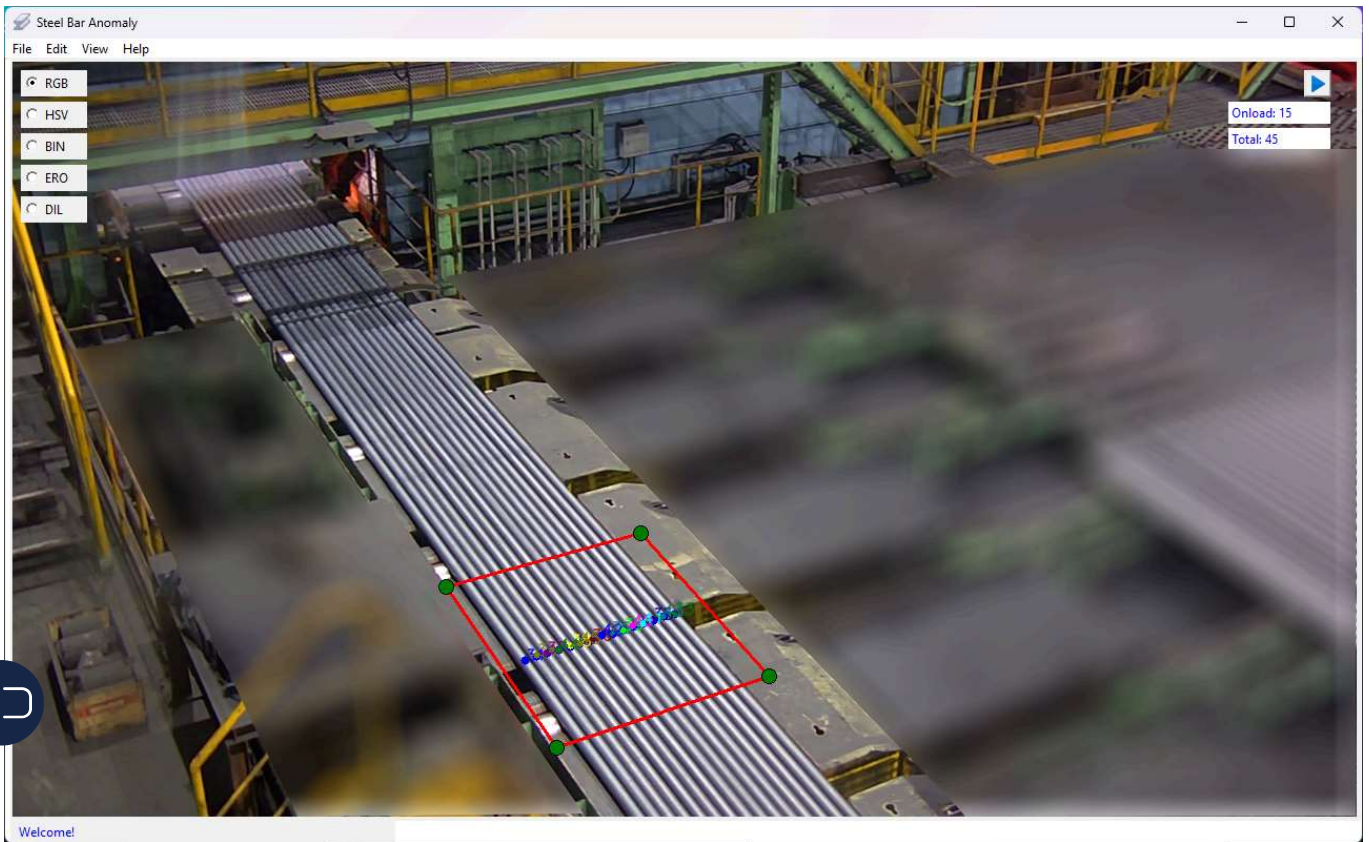
§ My Contribution

I played a role in developing the Rebar Tracking application, focusing on Python programming with tkinter and integrating PostgreSQL for data handling.

§ Monitoring & Recording

Real-time monitoring of rebar movement using live video feeds from factory cameras.





✉ thanhdat6716@gmail.com

☎ +82-10-4409-6699

in thanhdat6716